USSN 10/767,862

PATENTS • DESIGNS
The Patent Office
COPYRIGHT • TRADE MARKS

CHARTER MARK

INVESTOR IN PEOPLE

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) f the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the omptroller-General, hereby certify that annexed hereto is a true copy of the documents as iginally filed in connection with the patent application identified therein.

ccordance with the Patents (Companies Re-registration) Rules 1982, if a company named is certificate and any accompanying documents has re-registered under the Companies Act 0 with the same name as that with which it was registered immediately before re-stration save for the substitution as, or inclusion as, the last part of the name of the words lic limited company" or their equivalents in Welsh, references to the name of the company is certificate and any accompanying documents shall be treated as references to the name which it is so re-registered.

ordance with the rules, the words "public limited company" may be replaced by p.l.c., .L.C. or PLC.

stration under the Companies Act does not constitute a new legal entity but merely the company to certain additional company law rules.

Signed

Dated 4 February 2004

CERTIFIED COPY OF
PRIORITY DOCUMENT

Patents Form 1/77

Paten[illegible] 1977
(Rule [illegible]

The Patent Office

Cardiff Road
Newport
South Wales
NP10 8QQ

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

1. Your reference — 200206116-3 GB3

2. Patent application number *(The Patent Office will fill in this part)* — 29 MAY 2003 — 0312229.8

3. Full name, address and postcode of the or of each applicant *(underline all surnames)*

Hewlett-Packard Development Company, L.P.
20555 S.H. 249
Houston, TX 77070
USA

Patents ADP number *(if you know it)* 08557886001

If the applicant is a corporate body, give the country/state of its incorporation

4. Title of the invention — Privacy Management of Personal Data

5. Name of your agent *(if you have one)*

"Address for service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*

Robert F. Squibbs
Hewlett-Packard Ltd, IP Section
Filton Road, Stoke Gifford
Bristol BS34 8QZ

Patents ADP number *(if you know it)* 07928187001

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number

Country	Priority application number *(if you know it)*	Date of filing *(day / month / year)*
GB	0302270.4	31/01/2003
GB	0304049.0	24/02/2003

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application	Date of filing *(day / month / year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*
a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.
See note (d))

Yes

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	
Description	20
Claim*(s)*	11
Abstract	1
Drawing*(s)*	4 + 4

10. If you are also filing any of the following, state how many against each item.

Priority documents	
Translations of priority documents	
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	
Request for preliminary examination and search *(Patents Form 9/77)*	
Request for substantive examination *(Patents Form 10/77)*	
Any other documents *(please specify)*	Form 23/77 (x2) and Fee Sheet

11. I/We request the grant of a patent on the basis of this application.

Signature R. F. Squibbs
Robert F. Squibbs

Date 23 /5/2003

12. Name and daytime telephone number of person to contact in the United Kingdom

Tony Judd Tel: 0117-312-8026

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

a) If you need help to fill in this form or you have any questions, please contact the Patent Office on 08459 500505.

b) Write your answers in capital letters using black ink or you may type them.

c) If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.

d) If you have answered 'Yes' Patents Form 7/77 will need to be filed.

e) Once you have filled in the form you must remember to sign and date it.

f) For details of the fee and ways to pay please contact the Patent Office.

Privacy Management of Personal Data

Field of the Invention

The present invention relates to privacy management of personal data.

As used herein, the term "personal data" is intended to include data such as identity data and profile data (for example, preference data and financial data) of a party to which the data relates, whether that party is a natural or legal party. Furthermore, references to the "owner" of the personal data means the party responsible for its disclosure, whether the party is the subject of the data or a proxy for that party.

Background of the Invention

Digital identities and profiles of parties are becoming more and more relevant for enabling Internet transactions and interactions among citizens, service providers, enterprises and government institutions. For example, in an e-commerce scenario, a person initially provides their digital identity and profile information to an e-commerce site in order to access their services. After the user logs in and interacts with these services: it might happen that interaction with other web sites or organisations is needed to carry out a service. The user might be conscious of this or this might take place behind the scene, for example due to fact that the e-commerce site interacts with partners and suppliers. The e-commerce sites may or may not have prior agreements with these third parties or may or may not belong to the same web of trust.

In general users have little understanding or knowledge of the privacy laws and legislation that regulate the management of their information. The privacy and data protection laws that regulate this area are hard to enforce or monitor, especially when private information is spread across organisations and national' boundaries. People perceive and address the related security and privacy issues in different ways, ranging from completely ignoring them (and indiscriminately disclosing their personal data), to being so concerned as to refrain from using any Internet applications. It is also frequently the case that users do not bother to read long lists of terms and conditions concerning privacy and confidentiality because they cannot understand them or do not have the time to do so. Thus, whilst users are often asked to grant authority to web sites to electronically manage their information, in many cases the user doesn't consider the implications of such a request and simply chooses the easiest way forward to obtaining the service they want.

Little has been done so far to allow the explicit management and enforcement of privacy policies by directly involving users (or entities acting on their behalf) especially in a context of multiparty interactions. Users have a lack of control over their personal information, especially after its initial disclosure. In addition, third parties (such as delegates, e-commerce sites or enterprises) have lack of control over the confidential information they manage on behalf of their customers, in particular when they disclose it to external entities, during transactions or interactions.

Privacy management solutions can play a key role in protecting identities and profiles, enforcing good management practices and helping to detect criminal activities and support forensic analysis. However, for such solution to succeed, they need to simplify users' experience so that people can feel they are in control of their personal data and that this data is managed in an accountable way. If people are not willing to be involved in the active protection and management of their digital assets, trusted third parties could do this on their behalf and could provide people with easy-to-use tools to monitor and keep the situation under control.

Mechanisms such as proposed by W3C allow users to define simple privacy policies but this is only meaningful for point-to-point interactions (see: "The Platform for privacy preferences 1.0 specification (P3P 1.0)." http://www.w3.org/tr/p3p - W3C Proposed Recommendation – 2002)

Solutions based on federated identity management have also been implemented (such as Microsoft Passport) but, at least currently, rely on a closed web of trust. Identity providers must be part of trusted clubs and be compliant with predefined privacy policies. This approach limits scalability and flexibility of the allowed interactions and transactions.

A more fine-grained control over the privacy of personal data has been described in the papers:

- G. karjoth, M. Hunter – A Privacy Policy Model for Enterprises, IBM Research, Zurich – 15th IEEE Computer Foundations Workshop – June 2002
- G. karjoth, M. Schunter, M. Waidner – Platform for Enterprise Privacy Practices: Privacy-enabled Management of Customer Data – 2nd Workshop on Privacy Enhancing Technologies, Lecture Notes in Computer Science, Springer Verlang - 2002

In the first of these papers the authors define a privacy control language that includes user consent, obligations and distributed administration. In the second paper, the authors describe a platform for enterprise privacy practices (E-P3P) and introduce the "sticky policy" paradigm and mechanisms for enterprise privacy enforcement. Sticky policies are policies that are strictly associated with a user's data and drive access control decisions and privacy enforcement. The papers do not, however, describe how the strong associations between policies and confidential data are enforced, especially across enterprise boundaries. Users still need to trust the enterprise when disclosing their data. Leakage of personal and confidential information might happen, despite data protection laws and privacy policies, because of lack of security, dishonesty of some of the involved intermediaries and the complexity of the overall systems.

Furthermore, many of the current privacy mechanisms introduce an overhead in terms of usage of digital certificates at the user site (where data is encrypted) and complexity when dealing with dynamic metadata (policies) associated with the encrypted data

It is an object of the present invention to provide an improved way of effecting privacy management for personal data.

The present invention is in part based on the appreciation that Identifier-Based Encryption (IBE) has certain properties than can be adapted for use in privacy management.

Identifier-Based Encryption (IBE) is an emerging cryptographic schema. In this schema (see Figure 1 of the accompanying drawings), a data provider 10 encrypts payload data 13 using an encryption key string 14 and public data 15 provided by a trusted authority12; the data provider 10 then provides the encrypted payload data to a recipient 11 who decrypts it using a decryption key 16 provided by the trust authority together with the latter's public data. The trusted authority's public data is derived by the authority using private data 17 and a one-way function 18. Important features of the IBE schema are that any kind of string (including a name, a role, etc.) can be used as an encryption key string 14, and that the generation of the decryption key 16 is effected by the trust authority (process 19) using the encryption key string 14 and its private data 17, enabling the generation of the decryption key 16 to be postponed until needed for decryption. Because the encryption key string frequently contains data identifying the intended recipient (for example, by a required characteristic), the encryption key string is also known as the identifier string.

A number of IBE algorithms are known, one of which is the "Quadratic Residuosity" (QR) method described in the paper: "An Identity Based Encryption Scheme based on Quadratic Residues". C. Cocks Communications-Electronics Security Group (CESG), UK. http://www.cesg.gov.uk/technology/id-pkc/media/ciren.pdf - 2001. A brief description of this form of IBE is given below.

In the QR method, the trust authority's public data 15 comprises a value N that is a product of two random prime numbers p and q, where the values of p and q are the private data 17 of the trust authority 12. The values of p and q should ideally be in the range of 2^{511} and 2^{512} and should both satisfy the equation: $p, q \equiv 3 \bmod 4$. However, p and q must not have the same value. Also provided is a hash function # which when applied to a string returns a value in the range 0 to N-1.

Each bit m of the user's payload data 13 is then encrypted as follows:

- The data provider 10 generates random numbers t_+ (where t_+ is an integer in the range [0, 2^N]) until a value of t_+ is found that satisfies the equation *jacobi*(t_+,*N*)=m, where m has a value of -1 or 1 depending on whether the corresponding bit of the user's data is 0 or 1 respectively. (As is well known, the *jacobi* function is such that where $x^2 \equiv \# \bmod N$ the jacobi (#, N) = -1 if x does not exist, and = 1 if x does exist). The data provider 10 then computes the value:

$$s_+ \equiv (t_+ + \#(encryption\_keystring)/t_+)\bmod N$$

where s_+ corresponds to the encrypted value of the bit m concerned.

- Since #(encryption_keystring) may be non-square, the data provider additionally generates additional random numbers t_- (integers in the range $[0, 2^N)$) until one is found that satisfies the equation $jacobi(t_-, N) = m$. The data provider 10 then computes the value:

  $$s_- \equiv (t_- - \#(encryption\_keystring)/t_-)\bmod N$$

  as the encrypted value of the bit m concerned.

The encrypted values s_+ and s_- for each bit m of the user's data are then made available to the intended recipient 11, for example via e-mail or by being placed in a electronic public area; the identity of the trust authority 12 and the encryption key string 14 will generally also be made available in the same way.

The encryption key string 14 is passed to the trust authority 12 by any suitable means; for example, the recipient 11 may pass it to the trust authority or some other route is used - indeed, the trust authority may have initially provided the encryption key string. The trust authority 12 determines the associated private key B by solving the equation :

$$B^2 \equiv \#(encryption\_keystring)\bmod N \qquad \text{(“positive” solution)}$$

If a value of B does not exist, then there is a value of B that is satisfied by the equation:

$$B^2 \equiv -\#(encryption\_keystring)\bmod N \qquad \text{(“negative” solution)}$$

As N is a product of two prime numbers p, q it would be extremely difficult for any one to calculate the decryption key B with only knowledge of the encryption key string and N. However, as the trust authority 12 has knowledge of p and q (i.e. two prime numbers) it is relatively straightforward for the trust authority 12 to calculate B.

Any change to the encryption key string 14 will result in a decryption key 16 that will not decrypt the payload data 13 correctly. Therefore, the intended recipient 11 cannot alter the encryption key string before supplying it to the trust authority 12.

The trust authority 12 sends the decryption key to the data recipient 11 along with an indication of whether this is the "positive" or "negative" solution for B.

If the "positive" solution for the decryption key has been provided, the recipient 11 can now recover each bit m of the payload data 13 using:

$$m = jacobi(s_{+}+2B,N)$$

If the "negative" solution for the decryption key B has been provided, the recipient 11 recovers each bit m using:

$$m = jacobi(s_{-}+2B,N)$$

Whilst in the foregoing example, the encryption key string has been used directly in the QR IBE algorithm, it is also possible to use in the encryption process a derivative of the encryption key string, this derivative being formed, for example, by using a predetermined hash function. In this case, the entity generating the decryption key can still simply be supplied with the encryption key string provided it knows the predetermined function used to form the derivative of the encryption key string (in fact, this is equivalent to using a variant of stated the QR IBE algorithm in which the predetermined function is applied to the encryption key string wherever the latter appears). Where the decryption-key generating entity does not need to access the contents of the original encryption key string, then it need only be provided with the derivative of the encryption key string used during the encryption process. In the following description, where the term "encryption key" is used, this is intended to refer to the form of the encryption key string used in the stated version of IBE algorithm concerned whether this is the unprocessed encryption key string or a derivative formed by subjecting the encryption key string to predetermined processing.

Other IBE algorithms are known such as the use of Weil or Tate pairings – see, for example: D. Boneh, M. Franklin – "Identity-based Encryption from the Weil Pairing" in

Advances in Cryptology - CRYPTO 2001, LNCS 2139, pp. 213-229, Springer-Verlag, 2001. IBE algorithms based on the Weil or Tate pairings are usually described in terms of there being an IBE encryption key that is derived in a predetermined manner from an encryption key string (though it would be possible to re-state the algorithms such that the encryption key string formed the encryption key to be plugged into the algorithm).

Summary of the Invention

In general terms, the present invention involves using a privacy policy as an IBE encryption key string for the personal data to which it relates thereby tightly associating the policy and data and requiring the policy to be disclosed, unaltered, to the trust authority who has the ability to provide the decryption key. The trust authority then has the responsibility of ensuring that the policy conditions have been satisfied before it releases the decryption key. No secret needs to be generated and exchanged between users and the receivers of confidential information.

More particularly, according to one aspect of the present invention, there is provided a privacy management method, comprising:

first operations, effected by an owner of personal data, comprising encrypting the personal data and providing the encrypted data to a recipient, the encryption process using both:

- an encryption key formed using at least policy data indicative of conditions to be satisfied before access is given to said personal data; and
- public data provided by a trusted party and derived thereby using private data;

second operations, effected by the trusted party, comprising using the encryption key and said private data to determine a decryption key, and outputting this decryption key; at least one of these second operations only being effected after a further second operation has checked that said conditions are satisfied regarding said recipient.

The conditions to be satisfied may relate to the authenticity of the recipient, the security rating of the computing platform used by the recipient, a "use-before" date for the policy or data, etc; a condition may also be that the trusted party communicate with the owner of the personal data either by way of a simple notification or to get permission to deliver the decryption key.

The trusted party preferably keeps an audit record of each decryption key it delivers and each failed request for a key.

According to another aspect of the present invention, there is provided a privacy management system comprising first, second and third computing entities, wherein:

- the first computing entity comprises: a data store holding personal data; an encryption unit for encrypting the personal data using both an encryption key formed using at least policy data indicative of conditions to be satisfied before access is given to said personal data, and public data provided by the second computing entity; and a communications interface for providing the encrypted data to the third computing entity;
- the second computing entity comprises a data store holding private data; a communications interface for receiving the encryption key and for providing a corresponding decryption key to the third computing entity; a decryption-key determination unit for using the private data and the received encryption key to determine the corresponding decryption key for decrypting the encrypted data; and a condition-checking arrangement for ensuring that the decryption key is only determined, or only provided to the third computing entity, after the conditions in said policy data have been satisfied in respect of the third computing entity.

According to a further aspect of the present invention, there is provided a computing entity arranged to act as a trusted party, the computing entity comprising:

- a data store holding private data;
- a communications interface for receiving an encryption key and for outputting a corresponding decryption key to a requesting entity; the encryption key being formed using at least policy data indicative of conditions to be satisfied before access is given to data encrypted with the key;
- a decryption-key determination unit for using the private data and a received encryption key to determine a corresponding decryption key for decrypting data

encrypted using the encryption key and public data derived using said private data; and

- a condition-checking arrangement for ensuring that the decryption key is only determined, or only output via the communications interface, upon the conditions in said policy data being satisfied in respect of the requesting entity.

Brief Description of the Drawings

Embodiments of the invention will now be described, by way of non-limiting example, with reference to the accompanying diagrammatic drawings, in which:

. Figure 1 is a diagram illustrating the operation of a prior art encryption schema known as Identifier-Based Encryption;

. Figure 2 is a diagram of an embodiment of the present invention;

. Figure 3 shows an XML-format message comprising a privacy policy and data encrypted using the policy as the encryption key according to the IBE schema; and

. Figure 4 is a diagram of a policy hierarchy.

Best Mode of Carrying Out the Invention

Figure 2 illustrates a privacy management system in which a data-owner computing entity 20 is arranged to encrypt personal data and send it to a data-recipient computing entity 30 which then requests a decryption key from a trust authority computing entity 40 and, on receipt of the key, decrypts and uses the personal data. The computing entities 20,30 and 40 inter-communicate, for example, via the internet or other computer network though it is also possible that two or all three entities actually reside on the same computing platform.

The system employs Identifier-Based Encryption with the computing entities 20, 30 and 40 having the roles of the data provider 10, data recipient 11 and trusted authority 12 of the Figure 1 IBE arrangement. The IBE algorithm used is, for example, the QR algorithm described above with respect to Figure 1. The encryption key used to encrypt the personal data is a privacy / disclosure policy setting out conditions that must be satisfied before access is given to the personal data. This policy and the related personal data is made available in any suitable manner (including by direct peer-to-peer communication, by e-mail, or by posting on a public site) to the data recipient entity 30. In the Figure 2 example, the policy and related personal data are depicted as being sent in a data package 25 directly to the data recipient entity 30 (see arrow 50). On receipt, the entity 30 forwards the policy to the trust authority entity 40 with a request for a decryption key (see arrow 51). The trust authority entity 40 is then responsible for ensuring that all the conditions of the policy have been met either before it generates the decryption key, or before it supplies the decryption key to the recipient entity 30 (see arrow 53). One possible condition involves the trust authority entity 40 communicating with the owner entity 20 (see arrow 52) either simply to notify the latter or to obtain authorisation to proceed with the provision of the decryption key to the recipient entity 30. Advantageously, the trust authority entity keeps an auditable record of its interactions with the recipient entity. The trust authority entity will typically serve multiple data recipient entities in respect of data from multiple data owner entities.

More particularly, the data-owner 20 entity comprises a data store 21 for holding personal data and related disclosure policies, a browser 22 providing a user interface for managing interaction with the recipient entity 30, and a communications module 24 for communicating with the other entities 30, 40. The browser 22 has a plug-in 23 that provides the IBE functionality needed by the entity 20, this plug-in 22 being provided, for example, by the trust authority entity 40. Where the QR IBE method is being used, the plug-in thus contains the public data N and the hash function # together with program code for encrypting data using N and an encryption key string formed by the disclosure policy relevant to the data concerned.

Preferably, the personal data is divided into multiple components each with its own disclosure policy whereby different conditions can be set on different items of personal data. The data package 25 out by the entity 20 may include one or more personal-data components and their related policies.

With respect to the or each policy, such a policy can include conditions relating to:

- the strength of cryptographic methods to be employed in authenticating the identity of recipient before the decryption key is provided to the latter.

- the expiry date of the policy or of the personal data, the trusted authority being arranged not to the decryption key when the expiry date has passed.
- a security parameter of a computing platform being used by the recipient.
- an action to be performed by the trust authority entity such as communicating with the owner, the trusted party effecting this communication before providing the decryption key to said recipient.

Other types of condition are also possible.

The policies can be expressed in any suitable language, for example XML. Figure 3 shows an example data package 25 in XML format for one data component (attribute 1); as can be seen the package comprises a policy section 26 and an encrypted data section 27 (the dashed lines simply being included to delimit these sections for the purpose of clarity).

The policy illustrated in the policy section 26 of the Figure 3 data package 25 comprises:

- An encrypted "identifier" of owner (see "owner details" tag). This can be any information, including the owner's e-mail address, URL, etc. In this example, a "reference name" (a pseudonym, for example) has been used as an IBE encryption key to encrypt this information. Only the competent trust authority entity 40 will be able to retrieve the owner's identifier (and use it, for example, to notify the owner of a disclosure or ask for an authorization).
- The name of the attached confidential attribute (see "target" tag);
- An expiration date for the policy or associated attribute data (see ""validity" tag): after this date the trust authority entity 40 is required not to issue the decryption key;
- Policy conditions divided into constraints and actions: the constrains require the recipient entity 30 to strongly authenticate itself to the trust authority entity 40, and specify the usage of the attribute. The action condition requires the trust authority entity to notify the user of a disclosure.

Any kind of condition can be added, as long as the trust authority and the recipient entity can understand its semantic. The format adopted for the policy in its form included in the data package 25 and its form used as the IBE encryption key string need not be the same provided the forms used are known to the entities who have a need to know.

Considering next the data recipient entity 30, this comprises a credentials database 31, an IBE decryption module 32, a policy engine 33 and a communications module for communicating with the entities 20 and 30. On receipt of the data package 25, the policy engine 33 programmatically interprets the associated disclosure policies in order to determine what information (including authentication credentials, business related information, company/individual policy related to data disclosure, usage and storage, software state, platform configuration etc.) it will need to provide to the trust authority entity 40. The policy engine 33 is then responsible for sending to the entity 40, in respect of each encrypted personal-data component, a request for the decryption key, this request being accompanied by the relevant policy and the information which the engine believes is required from it to satisfy the conditions in the policy.

The receiving entity is thus explicitly aware of the conditions put on access to the encrypted data.

The trust authority entity 40 comprises a data store 41, a decryption key generation module 42, a policy engine 43 (different in functionality to that of the entity 30), an audit data module 44, and a communications module 46 for communicating with entities 20 and 30. On receiving a request for a decryption key from the entity 30, the policy engine 43 of the trust authority programmatically interprets the conditions in the associated policy and determines whether the information provided by the entity 30 in the request satisfies all the conditions in the policy that are satisfiable by the entity 30. The policy engine 43 may determine that the information given is inadequate and may send back a query to the entity for further information. Certain conditions in the policy may not rely on information from the entity 30 to be satisfied; one such condition is an action condition requiring the entity 40 to notify the data-owner entity 20 or to seek its explicit authorisation for release of the decryption key concerned.

If and when the policy engine 43 is satisfied that all policy conditions have been met, it causes the key generation module 42 to generate the required decryption key from the policy (acting as the corresponding encryption key) and the private data (the value N in the case of the QR IBE method) securely stored in store 41. The decryption key is then sent back to the entity 30. However, if one or more of the policy conditions is not satisfied, the entity 40 notifies the entity 30 accordingly and does not generate or output the requested decryption key.

It will be appreciated that rather than the entity 30 providing the information required for satisfaction of policy conditions in the decryption-key request, this information can be requested by the entity 40 as required to satisfy each condition as it is inspected by the policy engine 43. Furthermore, the decryption key can be generated at the same time as, or even before, the policy conditions are checked; in this case, the decryption key is not, however, released until the conditions are all found to be satisfied.

Whether or not a decryption-key request is successful, the audit data module 44 generates an audit record 47 comprising the identities of the entities 20 and 30, the personal-data component concerned and the information used to satisfy – or failing to satisfy – each policy condition. This audit record 47 is stored in store 41 to provide an audit trail regarding the disclosure of personal data and attempted accesses to it; this audit trail can be used latter as evidence for future contentions or forensic analysis.

Thus, if the recipient entity 30 discloses data in a way that is not allowed by the policies, there is an audit trail at the trust authority entity 40 showing that the entity 30 knew about the policy. In case of identity or profile thefts, the audit information can be used to pin down a list of potential "offenders" and carry on forensic analysis. Enforcing the tracing and auditing of disclosures makes the information recipients more accountable.

The trust authority entity 40 is the most suitable place to implement tracing and auditing activities as data recipients 30 need to interact with the trust authority entity 40 to obtain an IBE decryption key.

It should be noted that once personal data has been disclosed to a recipient entity 30 and it is in clear text (at the recipient site), it can potentially be misused. However, the provision of audit information in described system facilitates the identification of the source of any abuses.

In the foregoing example of a data package 25 given with respect to Figure 3, only one data component and one associated policy is shown. However, it will be appreciated that the data package can contain multiple data components each with its own associated policy in which case the trust authority entity 40 is arranged to provide a corresponding number of decryption keys each subject to the satisfaction of the conditions in the corresponding policy. Of course, the same policy can be applied to multiple items of the personal data. Furthermore, it is possible to provide a set of policies where two or more policies can be used in combination to protect a particular item of personal data whilst a different combination of policies can be used to protect a different item of personal data.

Figure 4 depicts a set of policies organised as a tree-structured hierarchy with policy P1 forming the root (first level) node to apply to all data, policies P2.1, P2.2 and P2.3, forming second-level nodes, and policies P3.1 to P3.7 forming third-level nodes. Data items to be encrypted are associated with one or more of the nodes (as indicated by the rectangular boxes "D" and dashed lines in Figure 4). To encrypt any particular data item, either a "policy concatenation" or a "policy nesting" approach is applied, as explained below:

Policy Concatenation - with this approach, all the policies traversed from the root node to the node with which the data item concerned is associated, are concatenated (in their order of traversal or the reverse order), and the concatenated policies are then used as the encryption key for encrypting the data item.

Policy Nesting – with this approach, the policy of the node with which the data item concerned is associated, is used to encrypt the data item and the encrypted data item then becomes a data item associated with the parent node of the node just used. In their turn the data items of the parent node are encrypted (either individually, or all together) using the corresponding policy to become one or more data items for the node above, and so on. This approach requires encryption to be initiated from the bottom up (that is, starting at the leaf nodes)

In both cases, each policy may specify any appropriate trust authority though, in the "policy concatenation" approach, if the policies being concatenated specify different trust authorities, one is selected to be used for the concatenation.

In one example of a hierarchy of policies where "policy concatenation" is applied, the second-level policies are used as class policies that are to apply to respective different classes personal data items, and third-level policies are used as policies that are to apply to respective individual personal data items. In this case, items of personal data are only associated with the leaf (third-level) nodes so that every item of personal data is guarded by a combined policy made up of the concatenation of the root policy, the appropriate class (second level) policy and the appropriate individual (third level) policy; the combined policy forms the encryption key for the data item and is used by the trust authority to derive the corresponding decryption key (after all the relevant policy conditions are satisfied). With this particular example in which data items are only associated with leaf nodes, it is still possible to dispense with the application of policy conditions at any one or more levels simply by arranging for one or more policies to be empty.

It will be appreciated that whilst it is preferable for the lower level policies to be consistent with the higher level ones, this is not essential as rules can be applied by the trust authority entity to resolve any policy conflicts - for example, a higher level policy can be arranged to overrule lower level policies (or vice versa), or a more specific policy condition can be arranged to overrule a more general one.

Although in Figure 4, each data item "D" is shown as associated with a single node, it would also be possible to associate a data item with multiple nodes; this would be advantageous where different branches of the policy hierarchy related to different policy topics and it was desired to apply multiple topics to a data item. In this case, combining the policies encountered in traversing the hierarchy from its root to each node associated with a subject data item can be done in a number of different ways. For example, a "policy concatenation" approach can be applied to all such policies, possibly with the elimination of repeated policies (nodes traversed more than once). Another approach is to use "policy concatenation" for each traversal and then use each concatenated policy set to encrypt the data item in turn. Yet another approach would be to use "policy nesting" with each level in the hierarchy being taken in turn (from the bottom up) and the concerned policies at the same level each being used in turn for encryption.

To enable a multiparty transaction, the recipient entity 30 can be authorised (for example, in a policy condition) to pass the overall encrypted data or any encrypted component of it to a further party (or parties) who then must contact the trust authority for the decryption key; again the decryption key is only provided if the relevant policy conditions are satisfied in respect of this further party In passing on the received personal data, the recipient entity 30 may decide to further encrypt portions of this data by using additional policies and in this case the module 32 would be arranged to carry out both decryption and encryption operations. This further encryption performed by the entity 30 may be applied either to the decrypted personal data items from entity 20, or to the data items whilst still in their form as encrypted by the entity 20 (in which case, the policy or policies applied by the data-owner entity 20 can conveniently encompassed within the data encrypted by the recipient entity 30). The policies applied by the entity 30 are of its own choosing and, of course, may specify a different trust authority to that specified by the entity 20. A further entity receiving the encrypted data from the entity 30 must use the trust authority specified by the entity 30 to obtain the decryption key(s) for unlocking the encryption applied by the entity 30; if this unlocked data comprises data encrypted by entity 20 and the relevant policy, then the further entity must now use the trust authority specified by the entity 20 to obtain the decryption key(s) to finally gain access to the personal data provided by entity 20.

As indicated in the foregoing discussions of the use of policies in combination and the passing on of personal data by the recipient entity 30 to another party, multiple trust authorities may need to be involved in providing access to the transmitted personal data. Of most interest is the situation where the provider of a particular item of personal data encrypts that data item in such a way that multiple trust authorities need to be involved to enable a receiving party to access the data item. One reason for doing this is that different trust authorities may have different competencies; for example, one trust authority may be competent to check platform security whilst another might be competent in the field of privacy. One way of requiring the involvement of multiple trust authorities is to use the "policy nesting" approach described above. However, it is also possible for the data provider to encrypt the data item using a key based on public data from each of multiple trust authorities (such public data being derived from private data of each trust authority), decryption of the encrypted item only being possible by obtaining a corresponding sub-key from each trust authority involved. Further information about how multiple trust authorities can be used is given in:

L. Chen, K. Harrison, A. Moss, D. Soldera, N. P. Smart, "Certification of Public Keys within an Identity Based System", LNCS 2433, ed. G. Goos, J. Hartmanis and J. van Leeuwen, Proceedings of Information Security, pp. 332-333, 2002.

Advantageously, one or more of the conditions of a policy require that the recipient entity 30 is a trusted platform with trusted integrity-checking mechanisms 35 that the trust authority entity 40 is to utilize to check that the software state of this platform is conformant with the disclosure policies, and that the platform correctly implements defined privacy management mechanisms. Suitable trusted mechanisms are described in:

TCPA - Trusted Computing Platform Alliance Main Specification v1.1, www.trustedcomputing.org, 2001.

The presence of trusted integrity-checking mechanisms 35 in the recipient entity 30 also permits the latter to be checked out by the data owner before any personal data is sent; such a check may be an alternative to, or additional to, the trust authority checking the recipient entity (it may be desirable for checking to be done both by the data owner and the trust authority since the state of the recipient entity may change between when the encrypted personal data is sent to the recipient and when it decides to access the data).

Preferably, one or both the computing entities 20 and 40 are also trusted platforms with TCPA integrity-checking mechanisms 25 and 45 respectively. In this case, one or more of the following further checks can be carried out:

- the trust authority's computing platform to be checked out by the data owner to ensure that the trust authority will operate as expected;

- the trust authority's computing platform to be checked out by the recipient of the data to help the recipient decide whether the trust authority can be trusted with the information that the recipient needs to provide in order for the decryption key to be issued;
- where the data-owner's personal data is forwarded by the recipient entity 30 to another computing entity, then that further entity can check out the trust authority and, assuming that the further entity is itself provided with trusted integrity-monitoring mechanisms, the further entity can be checked out by the trust authority and the recipient entity 30;
- the data owner's computer platform can be checked out by the trust authority or by the recipient entity.

The integrity-checking mechanisms 35 provided at the recipient entity 30 (or at any other subsequent recipient of the personal data of data-owner 20) can be used to effectively enforce proper handling of the personal data it receives, by requiring that the software state of the entity 30 corresponds to the use of software that can be trusted to operate in a predetermined manner (for example, a Trusted Operating Systems (OSs) or software with known behaviour that is being run in the absence of subversive software). Thus, where a Trusted OS can be arranged not to pass on data tagged in a certain manner to another entity, the data-owner entity can ensure that a particular data item is not disclosed beyond the recipient entity by tagging the data item in the appropriate manner and setting a policy condition to be checked by the trust authority, that the recipient entity must be running the Trusted OS before the decryption key is generated/ provided to the entity 30.

Rather than the trust authority being separate from the data owner, the personal-data owner entity 20 can be arranged to run trust authority services itself in order to have first hand understanding of what happens to its information and make ultimate decisions about release of decryption keys. In this case, the personal-data owners can directly use a TCPA integrity challenge to check that the computing platform of the recipient has not been corrupted, before proceeding with the data disclosure. (It may be noted that where the owner entity and trust authority entity are combined, the so-called "public" data of the trust authority may not, in practice, be published outside of the combined entity; however, the term "private" is still apt to distinguish the data concerned from the private data of the trust authority).

It will be appreciated that many other variants are possible to the above described embodiments of the invention. For example, the recipient entity 30 may choose to cache a received decryption key to decrypt the data package 25 at a later date. Furthermore, in order to prevent the use of a decryption key in respect of more than one output of personal data by the entity 20, a nonce, i.e. a random number, can be incorporated into the policy at each transmission. This ensures that the encryption key is unique thereby ensuring that the corresponding decryption key will also be unique.

Rather than the trust authority supplying the decryption key directly to the data recipient entity directly, the trust authority could send the key to the data-owner entity for forwarding to the data recipient entity.

Since in the Figure 2 embodiment of the trust authority 40, an audit record is kept of the owner 20 and recipient 30 of a particular data item for which the recipient entity 30 has been provided the decryption key, if the trust authority 40 subsequently receives a request from a further entity for the decryption key for the same data item, the trust authority 40 can check whether the implied onward transmission of the data by the entity 30 may have breached a condition of the policy associated with the data item. For simplicity, the trust authority may assume that the data item had the same associated privacy policy when handled by the recipient 30 as when handled by the subsequently-requesting entity; in this case, the trust authority need only check the policy conditions in the later request to see if the recipient entity 30 had the right to pass on the data item. However, it is also possible for the trust authority 40 to record the policy under which the decryption key was released to the entity 30 and, in this case, the trust authority can checked the recorded policy for a condition preventing onward transmission. If a breach is indicated, then the trust authority 40 is preferably arranged not to release the decryption key and to log the event (it may also immediately notify the data owner). Of course, even if the data item was disclosed to the recipient entity 30 under a policy forbidding onward disclosure, it is possible for the later-requesting entity to have legitimately received the data item as the data item may have been provided to the later-requesting entity by a different party (such as the data owner) having the right to do so; care therefore needs to be taken as to how the trust authority carries out the policy compliance check just described. In fact, inappropriate refusal to supply a decryption key can be largely avoided by having the party making the request for the decryption key, indicate from whom it received the data item; this additional information enables the trust authority to determine what policy was applicable to the party passing on the data item to the requesting party. An alternative would be to uniquely number each usage of a policy by the data owner (for example, by including a usage serial number or a nonce in the policy) so that where a request is made for a decryption key that is accompanied by the policy used as the encryption key, it is simple matter for the trust authority to check its audit records for any previous requests regarding the same policy usage and thus determine any breaches of a non-disclosure condition of the policy.

Whilst in the foregoing embodiment, the encryption key string formed using the policy data has been used directly in the QR IBE algorithm (that is, as the 'encryption key') as set out in the introductory portion of the specification, it is also possible to use a derivative of the encryption key string as the encryption key as explained in the introductory portion; in this case, the encryption key string, rather than or as well as the encryption key, is passed to the trust authority 40 so that the contents of the encryption key string are visible to the trust authority.

It will be appreciated that instead of the QR IBE method, the above-described embodiments can be implemented using other, analogous, cryptographic methods such as IBE methods based on Weil or Tate pairings.

The above-described privacy management system can be used in any area of application including e-commerce, financial, government and enterprise areas.

CLAIMS

1. A privacy management method, comprising:

first operations, effected by an owner of personal data, comprising encrypting the personal data and providing the encrypted data to a recipient, the encryption process using both:

- an encryption key formed using at least policy data indicative of conditions to be satisfied before access is given to said personal data; and
- public data provided by a trusted party and derived thereby using private data;

second operations, effected by the trusted party, comprising using the encryption key and said private data to determine a decryption key, and outputting this decryption key; at least one of these second operations only being effected after a further second operation has checked that said conditions are satisfied regarding said recipient.

2. A method according to claim 1, wherein the first operations further comprise providing the encryption key to said recipient along with the encrypted data; the method further comprising intermediate operations in which the recipient provides the trusted party with the encryption key and requests the decryption key.

3. A method according to claim 1 or claim 2, wherein the first operations further comprise providing details of the trusted party to said recipient along with the encrypted data.

4. A method according to any one of claims 1 to 3, further comprising said recipient sending on the encrypted personal data to a further party, and the trusted party providing the decryption key to that further party only after said conditions have been satisfied in respect of that further party.

5. A method according to claim 1, wherein in said first operations multiple items of personal data are encrypted each using said public data and a respective encryption key formed using at least respective policy data; the encrypted multiple items being provided to said recipient; and wherein in the second operations the trusted party determines the decryption key for at least one encrypted item using the corresponding encryption key and

said private data, the or each determined decryption key only being provided to said recipient after the conditions in the corresponding policy data have been satisfied.

6. A method according to claim 5, further comprising said recipient sending on a selected subset of said multiple encrypted items of personal data to a further party; and the trusted party providing to that further party a decryption key for an encrypted item provided to that party, only after the conditions in the corresponding policy data have been satisfied in respect of said further party.

7. A method according to claim 1, wherein the data owner has a set of policies that form respective nodes in a policy hierarchy, and wherein in said first operations, multiple items of personal data are encrypted and provided to said recipient, each such data item being independently associated with at least one node of the policy hierarchy and being encrypted using said public data and policy data formed by a concatenation of the policies of the nodes traversed between the root of the hierarchy and the said at least one node with which the data item is associated.

8. A method according to claim 1, wherein the data owner has a set of policies that form respective nodes in a policy hierarchy, and wherein in said first operations, multiple items of personal data are encrypted and provided to said recipient, each such data item being independently associated with at least one node of the policy hierarchy and being encrypted by an iterative process in which:

- the data item is encrypted using said public data and policy data formed by the policy of the said at least one associated node,
- the encrypted data thus produced then becoming a data item associated with the parent node of the or each node formed by the policy just used for encryption.

9. A method according to claim 1, wherein in said first operations, multiple items of personal data are encrypted and provided to said recipient, at least two of these data items being encrypted using public data of different respective trusted parties whereby the recipient must obtain the appropriate decryption key from a different one of the trusted parties in dependence on which data item the recipient wishes to access.

10. A method according to claim 1, wherein in said first operations an item of personal data is first encrypted using a first policy and the public data of a first trusted party with the encrypted data being then further encrypted using a second policy and the public data of a second trusted party whereby the recipient must obtain decryption keys from the two trusted parties in order to access the data item.

11. A method according to claim 1, wherein in said first operations the personal data is encrypted using public data provided by multiple trusted parties, the second operations being carried out by each of said multiple trusted parties to provide a respective decryption sub-key whereby to enable the recipient to decrypt the encrypted personal data by the combined use of the sub-keys from each trust authority; each trusted party ensuring that policy conditions for which it is competent have been satisfied before generating and/or outputting the corresponding sub-key.

12. A method according to claim 1, wherein the trusted party makes an audit record of each provision of a decryption key by the trusted party.

13. A method according to claim 12, wherein said audit record further comprises information about when a decryption key is not provided because a related policy condition has not been satisfied, this information including information about the condition failure.

14. A method according to claim 12, wherein the trusted party on receiving a request from a party for a decryption key in respect of a particular item of data, checks its audit records to ascertain whether the decryption key for that item has previously been provided to a different party, and if so, whether the policy associated with the data item permitted onward disclosure.

15. A method according to claim 14, wherein the trusted party, on determining that the decryption key for the data item was previously provided under a policy of no onward disclosure, refuses to provide the decryption key to the requesting party.

16. A method according to claim 1, wherein the first and second operations are repeated multiple times for the same or different personal data owned by the same or different personal-data owners and provided to the same or different recipients.

17. A method according to claim 16, wherein the trusted party makes an audit record of each provision of a decryption key by the trusted party.

18. A method according to claim 17, wherein said audit record comprises the identity of the personal data, personal-data owner and recipient concerned.

19. A method according to claim 17 or claim 18, wherein said audit record further comprises information about when a decryption key is not provided because a related policy condition has not been satisfied, this information including information about the condition failure.

20. A method according to claim 17, wherein the trusted party on receiving a request from a party for a decryption key in respect of a particular item of data, checks its audit records to ascertain whether the decryption key for that item has previously been provided to a different party, and if so, whether the policy associated with the data item permitted onward disclosure.

21. A method according to claim 20, wherein the trusted party, on determining that the decryption key for the data item was previously provided under a policy of no onward disclosure, refuses to provide the decryption key to the requesting party.

22. A method according to claim 1, wherein a said policy condition relates to the strength of cryptographic methods to be employed in authenticating the identity of the recipient before the decryption key is provided to the latter.

23. A method according to claim 1, wherein a said policy condition relates to the expiry date of the policy or of the personal data, the trusted party not providing the decryption key when the expiry date has passed.

24. A method according to claim 1, wherein a said policy condition relates to the trusted party communicating with the owner, the trusted party effecting this communication before providing the decryption key to said recipient.

25. A method according to claim 24, wherein the condition is that the trusted party obtain consent from the owner before providing the decryption key to said recipient.

26. A method according to claim 24, wherein contact details for the owner are contained in policy data in encrypted form, the contact details being encrypted using said public data of the trusted party and an encryption key formed by a data element also included in the policy data whereby the trusted party can form the corresponding decryption key and decrypt the encrypted contact details.

27. A method according to claim 1, wherein a said policy condition relates to a computing platform being used by the recipient being a trusted platform running software of predetermined functionality that cannot be subverted.

28. A method according to claim 1, wherein the trusted party checks that any party requesting the decryption key is using a trusted computing platform running software of predetermined functionality that cannot be subverted.

29. A method according to claim 1, wherein the data owner, before providing the encrypted data to the recipient, checks that the latter is using a trusted computing platform running software of predetermined functionality that cannot be subverted.

30. A method according to any one of claims 27 to 29, wherein the software being run by the computing entity of the recipient is arranged to prevent onward disclosure of data indicated in a predetermined manner, the data owner marking an item of personal data in this predetermined way before providing it to the recipient.

31. A method according to claim 1, wherein the data owner, before providing the encrypted data to the recipient, checks that the trust authority is using a trusted computing platform running software of predetermined functionality that cannot be subverted.

32. A method according to claim 1, wherein the recipient, before providing the trust authority with any data concerning itself for the purpose of satisfying a said condition, checks that the trusted party is using a trusted computing platform running software of predetermined functionality that cannot be subverted.

33. A method according to claim 1, wherein the recipient, before providing any personal data received from the data owner to another party, checks that the latter is using a trusted computing platform running software of predetermined functionality that cannot be subverted.

34. A method according to claim 1, wherein the owner of the personal data also serves as the trusted party.

35. A method according to claim 1, wherein said owner is acting as a proxy for a party to whom the personal data relates.

36. A method according to claim 1, wherein in the second operations the decryption key is not determined until after said conditions have been satisfied.

37. A privacy management system comprising first, second and third computing entities, wherein:

- the first computing entity comprises: a data store holding personal data; an encryption unit for encrypting the personal data using both an encryption key formed using at least policy data indicative of conditions to be satisfied before access is given to said personal data, and public data provided by the second computing entity; and a communications interface for providing the encrypted data to the third computing entity;

- the second computing entity comprises a data store holding private data; a communications interface for receiving the encryption key and for providing a corresponding decryption key to the third computing entity; a decryption-key determination unit for using the private data and the received encryption key to determine the corresponding decryption key for decrypting the encrypted data; and a condition-checking arrangement for ensuring that the decryption key is only determined, or only provided to the third computing entity, after the conditions in said policy data have been satisfied in respect of the third computing entity.

38. A system according to claim 37, wherein the first computing entity is arranged to provide the encryption key to the third computing entity along with the encrypted data; the third computing entity being arranged to request the decryption key from the second computing entity and provide it with the encryption key.

39. A system according to claim 37, further comprising a fourth computing entity, the third computing entity being arranged to send on the encrypted personal data to the fourth computing entity, and the second computing entity being arranged to provide the decryption key to the fourth computing entity only after said conditions have been satisfied in respect of that fourth computing entity.

40. A system according to claim 37, wherein the second computing entity is arranged to make an audit record of each provision of the decryption key by the second computing entity.

41. A system according to claim 40, wherein the second computing entity is arranged to include in the audit record, information about when a decryption key is not provided because a related policy condition has not been satisfied, this information including information about the condition failure.

42. A system according to claim 40, wherein the second computing entity is so arranged that upon receiving a request from a party for a decryption key in respect of a particular item of data, it checks its audit records to ascertain whether the decryption key for that item has previously been provided to a different party, and if so, whether the policy associated with the data item permitted onward disclosure.

43. A system according to claim 37, further comprising multiple first and third computing entities, the second computing entity being arranged to provide decryption keys for the third computing entities in respect of personal data encrypted by the first computing entities provided the corresponding policy conditions have been satisfied in each case.

44. A system according to claim 37, wherein the second computing entity is arranged to make an audit record of each provision of a decryption key by the second computing entity.

45. A system according to claim 44, wherein said audit record comprises the identity of the first and third computing entities concerned with each provision of a decryption key.

46. A system according to claim 44 or claim 45, wherein the second computing entity is arranged to include in the audit record, information about when a decryption key is not provided because a related policy condition has not been satisfied, this information including information about the condition failure.

47. A system according to claim 44, wherein the second computing entity is so arranged that upon receiving a request from a party for a decryption key in respect of a particular item of data, it checks its audit records to ascertain whether the decryption key for that item has previously been provided to a different party, and if so, whether the policy associated with the data item permitted onward disclosure.

48. A system according to claim 37, wherein a said policy condition relates to the second computing entity communicating with the first computing, the second computing entity being arranged to effect this communication before providing the decryption key to said third computing entity.

49. A system according to claim 48, wherein the condition is that the second computing entity obtain consent from the first computing entity before providing the decryption key to the third computing entity.

50. A system according to claim 48, wherein contact details of the first computing entity are included in said policy data in encrypted form, the contact details being encrypted using said public data and an encryption key formed by a data element also included in the policy data whereby the second computing entity can form the corresponding decryption key and decrypt the encrypted contact details.

51. A system according to claim 37, wherein a said policy condition relates to the third computing entity being a trusted platform running software of predetermined functionality that cannot be subverted.

52. A system according to claim 37, wherein the first and second computing entities are combined.

53. A computing entity arranged to act as a trusted party, the computing entity comprising:

- a data store holding private data;
- a communications interface for receiving an encryption key and for outputting a corresponding decryption key to a requesting entity; the encryption key being formed using at least policy data indicative of conditions to be satisfied before access is given to data encrypted with the key;
- a decryption-key determination unit for using the private data and a received encryption key to determine a corresponding decryption key for decrypting data encrypted using the encryption key and public data derived using said private data; and
- a condition-checking arrangement for ensuring that the decryption key is only determined, or only output via the communications interface, upon the conditions in said policy data being satisfied in respect of the requesting entity.

54. A computing entity according to claim 53, further comprising an audit-trail arrangement for making an audit record of each output of a decryption key to a requesting entity.

55. A computing entity according to claim 54, wherein the audit-trail arrangement is arranged to include in the audit record information about when a decryption key is not provided because a related policy condition has not been satisfied, this information including information about the condition failure.

56. A computing entity according to claim 54, in which the audit-trail arrangement is arranged, in response to the computing entity receiving a request from a party for a decryption key in respect of a particular item of data, to checks its audit records to ascertain whether the decryption key for that item has previously been provided to a different party, and if so, whether the policy associated with the data item permitted onward disclosure.

57. A computing entity according to claim 56, wherein the audit-trail arrangement is further arranged, on determining that the decryption key for the data item was previously provided under a policy of no onward disclosure, to block the generation and/or output of the decryption key.

58. A computing entity according to claim 53, wherein a said policy condition relates to the computing entity communicating with an owner of the encrypted data, the computing entity being arranged to effect this communication before generating and/or outputting the decryption key to the requesting entity.

59. A computing entity according to claim 58, wherein the condition is that the computing entity obtain consent from the owner of the encrypted data before providing the decryption key to the requesting entity.

60. A computing entity according to claim 53, wherein a said policy condition relates to the requesting entity being a trusted platform running software of predetermined functionality that cannot be subverted, the computing entity being arranged to communicate with the requesting entity to check this condition before generating and/or outputting the decryption key.

ABSTRACT

Privacy Management of Personal Data

When sending personal data to a recipient (30), the data owner (20) encrypts the data using both a public data item provided by a trusted party (40) and an encryption key formed using at least policy data indicative of conditions to be satisfied before access is given to the personal data. The encryption key is typically also provided to the recipient (30) along with the encrypted personal data. To decrypt the personal data, the recipient (30) sends the encryption key to the trusted party (40) with a request for the decryption key. The trusted party (40) determines the required decryption key using the encryption key and private data used in deriving its public data, and provides it to the requesting recipient (30). However, the decryption key is either not determined or not made available until the trusted party (40) is satisfied that the associated policy conditions have been met in respect of the recipient. One possible policy condition is that the trusted party (40) must first get confirmation from the owner (20) of the personal data before providing the decryption key. Preferably, the trusted party (40) keeps a record (47) of the provision of decryption keys

(Figure 2)

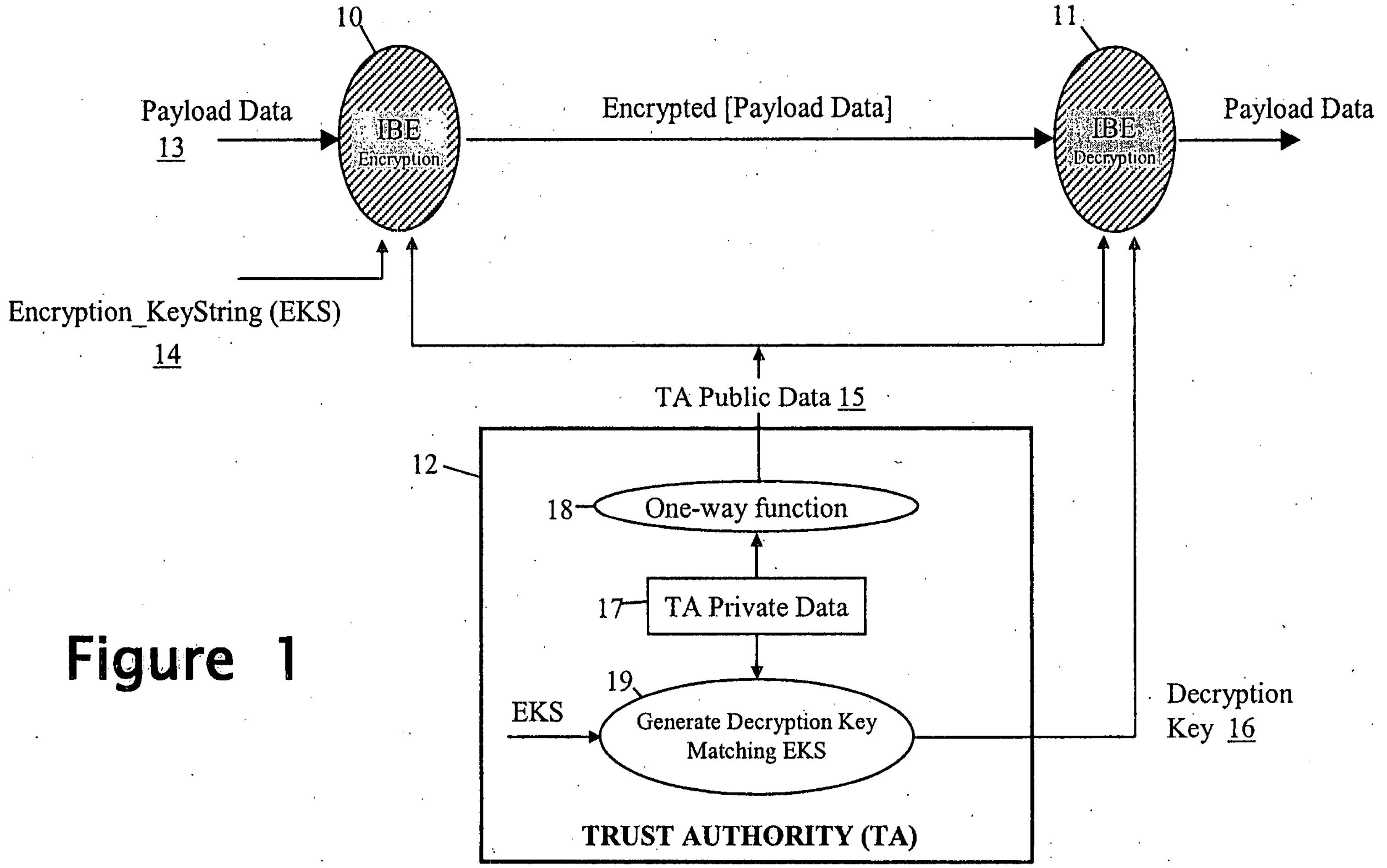

10
11
Payload Data
13
IBE
Encryption
Encrypted [Payload Data]
IBE
Decryption
Payload Data
Encryption_KeyString (EKS)
14
TA Public Data 15
12
18
One-way function
17
TA Private Data
19
EKS
Generate Decryption Key
Matching EKS
TRUST AUTHORITY (TA)
Decryption
Key 16

Figure 1

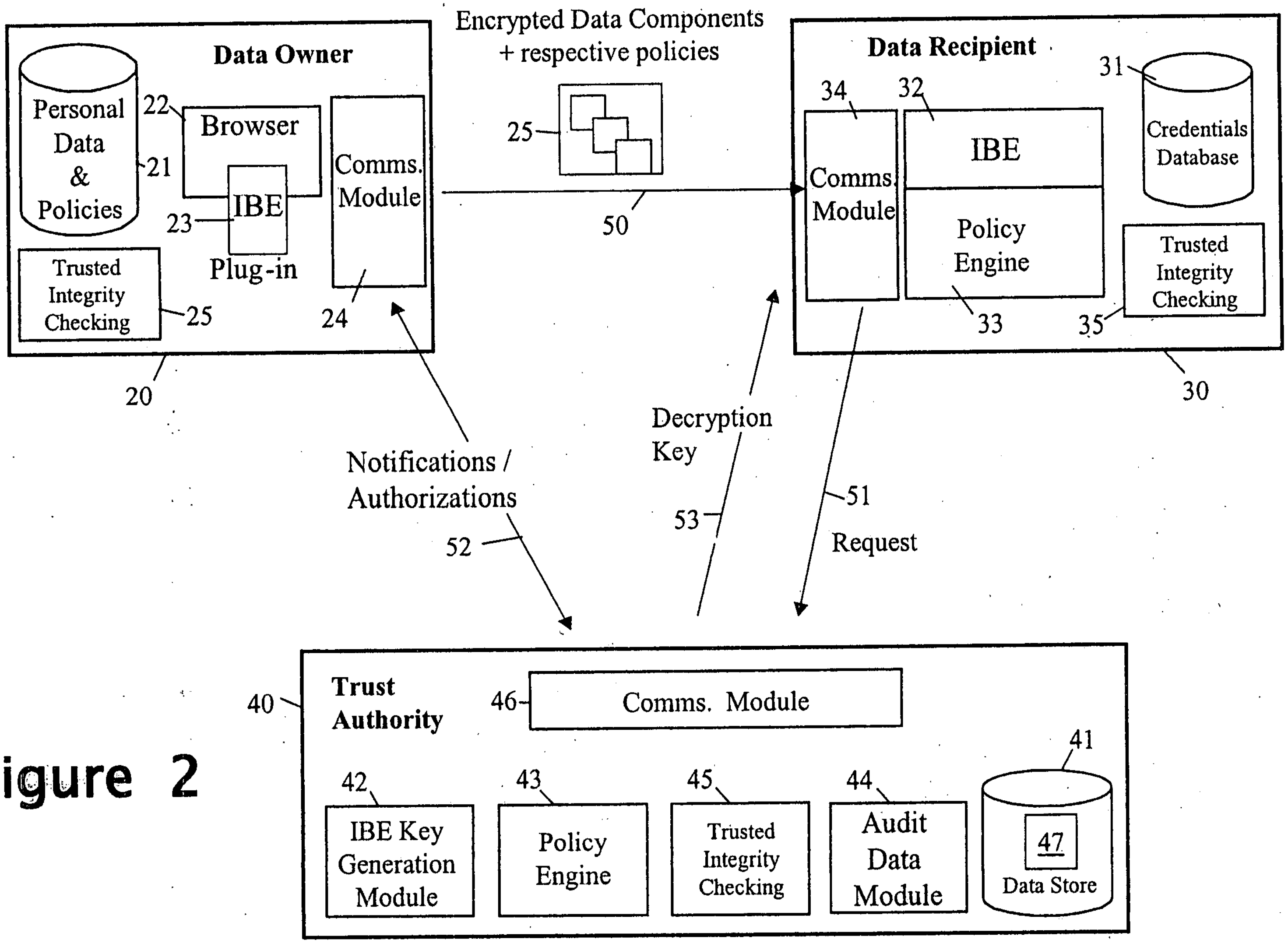
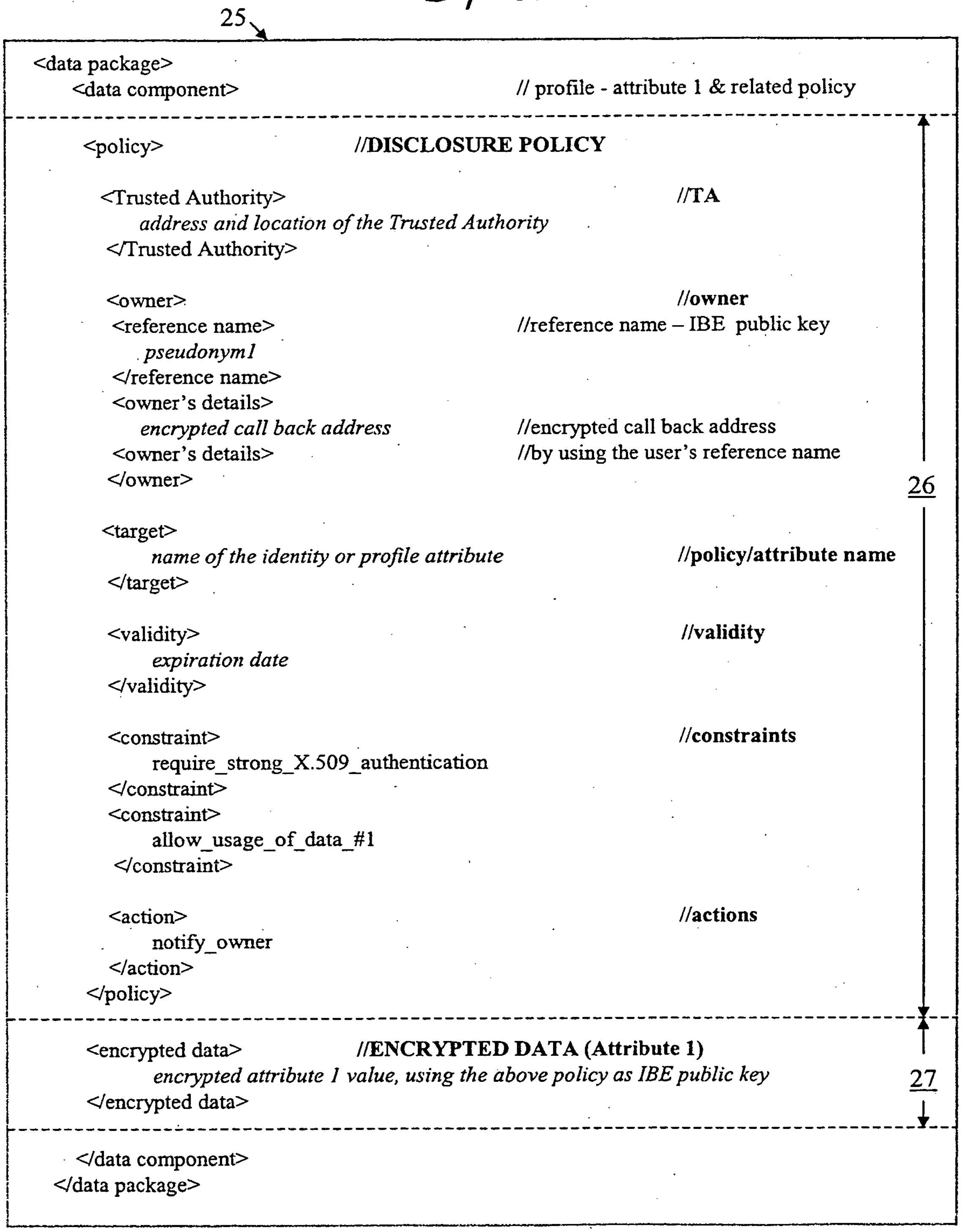

Data Owner
Personal Data & Policies
22
Browser
21
23
IBE
Plug-in
Comms. Module
Trusted Integrity Checking
25
24
20
Encrypted Data Components + respective policies
25
50
Data Recipient
34
32
31
Comms. Module
IBE
Policy Engine
Credentials Database
Trusted Integrity Checking
33
35
30
Notifications / Authorizations
52
Decryption Key
53
51
Request
Trust Authority
40
46
Comms. Module
41
42
43
45
44
IBE Key Generation Module
Policy Engine
Trusted Integrity Checking
Audit Data Module
47
Data Store

Figure 2



```
<data package>
    <data component>                              // profile - attribute 1 & related policy
-----------------------------------------------------------------------------------------------
      <policy>                  //DISCLOSURE POLICY

        <Trusted Authority>                                   //TA
            address and location of the Trusted Authority
        </Trusted Authority>

        <owner>                                               //owner
         <reference name>                         //reference name – IBE  public key
            pseudonym1
         </reference name>
         <owner's details>
            encrypted call back address           //encrypted call back address
         <owner's details>                        //by using the user's reference name
        </owner>

        <target>
            name of the identity or profile attribute         //policy/attribute name
        </target>

        <validity>                                            //validity
            expiration date
        </validity>

        <constraint>                                          //constraints
            require_strong_X.509_authentication
        </constraint>
        <constraint>
            allow_usage_of_data_#1
        </constraint>

        <action>                                              //actions
            notify_owner
        </action>
      </policy>
-----------------------------------------------------------------------------------------------
      <encrypted data>          //ENCRYPTED DATA (Attribute 1)
            encrypted attribute 1 value, using the above policy as IBE public key
      </encrypted data>
-----------------------------------------------------------------------------------------------
     </data component>
   </data package>
```

Figure 3

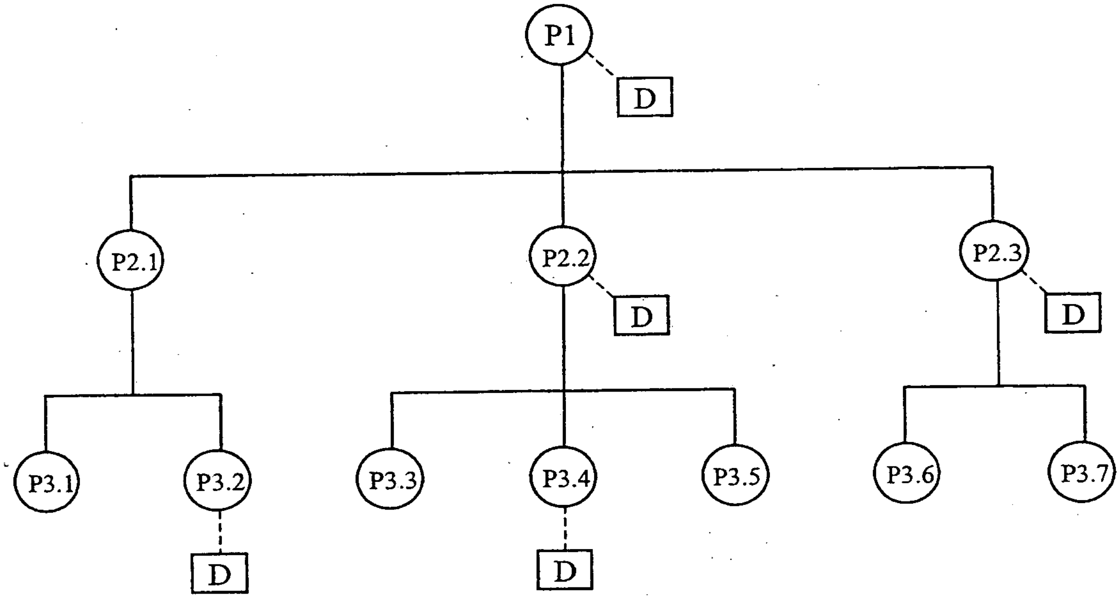


Figure 4